

Assignment #3

Naoaki Takatsu

Student ID: 015746144

CECS 326 Sec 05 5288 Operating Systems

Due 12, April 2018

Submitted 17, April 2018

Description:

shmp1.cpp:

This is the executable shared memory producer code. It will create 3 individual shared memory (child process) with each sharing an inventory of 15 seats. It will wait until the shared memory completes their tasks. When finished, it will detach and destroy the memory.

shmc1.cpp:

This is the shared memory consumer code. It will sell seats until the number of remaining seats become less than or equal to 0.

Problem Observed:

Consumers can “buy” seats even after the remaining number of seat hits 0. This occurs because the sleep system call in the sell\_seats() function in the shmc1.cpp can take long enough for other memory to access as well. Since the remaining seat count decrements after the sleep, other memory can pass through the if statement to sell the seats if they access it fast enough. This creates a race condition between the three memory. However, it can be solved by implementing mutual exclusion (boolean, semaphore, etc.).